ISSN 1840-4855
e-ISSN 2233-0046

Original scientific article
http://dx.doi.org/10.70102/afts.2025.1833.176

METAHEURISTIC-DRIVEN HYPERPARAMETER OPTIMIZATION FOR BERT IN SENTIMENT ANALYSIS

Alaa A. El-Demerdash[1*], Nahla B. Abdel- Hamid[2], Amira Y. Haikal[3]

[1]Computers and Control Systems Engineering, Faculty of Engineering, Mansoura University, Mansoura, Egypt. e-mail: alaa.abdelmohsen.eldemerdash@gmail.com, orcid: https://orcid.org/0000-0002-8056-6447*
[2]Computers and Control Systems Engineering, Faculty of Engineering, Mansoura University, Mansoura, Egypt. e-mail: nahla_bishri@mans.edu.eg, orcid: https://orcid.org/0000-0002-9515-5704
[3]Computers and Control Systems Engineering, Faculty of Engineering, Mansoura University, Mansoura, Egypt. e-mail: amirayh@mans.edu.eg, orcid: https://orcid.org/0000-0002-3097-8814

Received: May 12, 2025; Revised: July 31, 2025; Accepted: September 01, 2025; Published: October 30, 2025

SUMMARY

Sentiment analysis has come out as an important activity in natural language processing (NLP) applications whose data analysis is in high demand at present in the modern world. The BERT (Bidirectional Encoder Representations from Transformers) algorithm has proved to be extremely efficient when it comes to sentiment analysis tasks, and its potential is far exceeding that of conventional algorithms, unlocking their potential however would require fine tuning of their hyperparameters. It is quite a feat to optimise the BERT's various hyperparameters due to the complicated interaction between them (e.g. the learning rate, batch size, dropout rate, attention heads). In this paper, the Salp Swarm Algorithm (SSA) is used as a bio-inspired metaheuristic optimization technique to optimize the fine-tuning process. Through SSA's exceptionally efficient search capabilities in modelling multidimensional search space, BERT hyperparameters are optimized systematically to the sentiment classification tasks. A benchmark dataset for sentiment analysis (Sentiment140) is used to evaluate the proposed model. The novelty of the presented model is the fact that it dynamically adjusts its search behaviour in response to performance signals, thus it identifies better-performing parameter sets than conventional methods, leading to successful exploitation of the BERT algorithm that has produced high performing configurations. Extensive evaluations against 3 state-of-the-art search algorithms, namely manual tuning, grid search, and random search are conducted on the Sentiment140 benchmark dataset, demonstrating the superiority of the proposed SSA BERT optimization technique over state-of-the-art methods. The SSA-BERT model achieved a maximum accuracy of 96.4 percent, which is far better than manual tuning, grid search, and random search (65.0 percent, 69.5 percent and 72.0 percent respectively). It also performed better than other existing BERT models used in related literature, which showed accuracy levels between 46.4 and 75.7 percent in accordance with different benchmarks

Key words: *BERT, salp swarm algorithm, hyperparameter optimization, metaheuristics, natural language processing, sentiment analysis.*

INTRODUCTION

Sentiment analysis has become an increasingly important task within the domain of natural language processing (NLP) that is enhanced in a current data-centric environment. Its uses are diffused all over various fields such as monitoring customer satisfaction, analysing social networking, managing brand reputation, and mining opinion. Labelling sentiment into three distinct states, namely positive, negative and neutral automatically isn't only important academically but also holds significant value for businesses and organizations as it offers organizations the ability to understand and then react to public sentiment [1, 2].

Last few years, a strong shift in the section of NLP has become evident since the appearance of transformer-based models and BERT (Bidirectional Encoder Representations with Transformers). BERT introduced by [3], has introduced new standards in many NLP tasks, including sentiment analysis [39]. The strength of BERT lies in its novel architecture and pre-training model. BERT can be pre-trained with the help of significant amounts of unlabelled text data by using other methods, like masked language modelling and next sentence prediction, using the Transformer architecture. It is based on this pre-training that BERT creates comprehensive contextualized word descriptions that indicate subtle semantic and syntactic details [4].

BERT is a good backbone to several downstream NLP tasks such as sentiment analysis, although its effectiveness mainly lies on the fine-tuning process [5]. Fine-tuning modifies parameters of a pre-trained BERT model using datasets special to the tasks on which the model is customised [31]. Nevertheless, the success of such a fine-turning process lies especially heavily on the selection of hyperparameters [6]. The key hyperparameters that highly influence performance of BERT are the learning rate, the batch size, the dropout percentage and the number of attention heads [7].

Conventionally, the optimization of the hyper parameters has been approached through line of targeting the grid search and random search. The search in grid search is done in a methodical comb through an area of the hyperparameter space that is it chooses manually, whereas random search samples configurations in hyperparameter spaces as defined by given distributions [8]. Although these approaches are easy to apply, they can be ineffective and computationally expensive particularly, when dealing with large spaces having a substantial number of hyperparameters [9]. The drawbacks of these traditional methods also become particularly evident in the case of finetuning such advanced models as BERT, whose relationship between hyperparameters may be non-linear and complicated [33].

Due to the difficulty posed by the traditional approaches to hyperparameter optimization, there has been an attempt to develop more complex approaches to this optimization as seen with the introduction of meta-heuristic algorithms that are based on natural processes [10]. The benefits of these nature-based optimization techniques are many, such as their ability to effectively search large dimensional spaces, to evade local maxima, and to evaluate non-smooth objective functions [11].

The Salp Swarm Algorithm (SSA) has been verified and proven among researchers to be a reliable method in solving complex optimization problems [12]. First introduced by Mirjalili et al. in 2017, it emulates the way swarming salps form chains in the sea to enhance motion and foraging [13]. The SSA does this by simulating a population of so-called search agents (salps) moving around the optimization problem's search space while maintaining a balance between exploiting already discovered good ground and expansion by exploring new ground [14].

The SSA has certain properties which enables it to be a very suitable method in hyperparameter optimization problems. Its population-based approach enables multiple search agents to simultaneously explore the search space, while the leader-follower system creates efficient communication between them. Moreover, the SSA adjusts its search based on the current optimization progress which helps to prevent premature convergence to non-optimal solutions [15].

This paper applies the SSA as a proposed framework to optimize key hyperparameters in the BERT algorithm to achieve improved sentiment analysis. Two hyperparameters in BERT have proven to have a significant impact on its performance [16] which is why they're the focus of this paper's framework's optimization efforts.

The Sentiment140, a benchmark dataset in sentiment analysis [17, 18] was used to perform extensive experimentations to measure the effectiveness of the proposed SSA-based hyperparameter optimization method.

The main implications of this paper are:

- An alternative method of sentiment classification is proposed by integrating the Salp Swarm Algorithm (SSA) with BERT, and the aim is to optimise its hyperparameters. The model is applied on the benchmark sentiment analysis Sentiment140 dataset, to determine how well it performed on textual contexts.
- Several adaptations are introduced in the implementation process, including dynamic adjustment mechanisms within the SSA to better align with BERT's fine-tuning requirements. Additionally, the optimization workflow is customized to handle BERT-specific constraints and training dynamics.
- Comparative tests are performed against common optimization techniques such as manual tuning, grid search and random search.
- Statistical evaluation of the results is performed using t-test.

Section 2 reviews related work on the topic and the most relevant studies in sentiment analysis and BERT models as well as hyperparameter tuning strategies. Section 3 describes the planned methodology, i.e. how SSA will be applied to tune BERT hyperparameter and what experimental environment will be proposed. Section 4 contains and analyses our findings: how effective are SSA-tuned BERT models compared with baselines and what effect did various hyperparameters have. Lastly, Section 5 closes the document by giving a summary of all the findings and provide recommendation on how to go forward in the field.

RELATED WORK

Traditional Sentiment Analysis Techniques

Sentiment analysis is one of the basic elements of natural language processing which have advanced significantly over the years. In the early days of sentiment analysis, lexicon-based approaches and more traditional machine learning tools are relied upon heavily. Lexicon-based methods use pre-established lists of words that have known sentiment polarity to determine the overall sentiment of some writing. The techniques are simple and highly explicable and are therefore very handy when it comes to apportioning quick sentiments such as in social media tracking and customer feedback [1].

Catelli Conducted a comparative study of lexicon-grounded and BERT-grounded sentiment-analysis approaches in a setting where Italian is the language to be used. In their study, despite the advantages of dictionary-based approaches, they showed that such methods tend to have difficulty in the areas of handling the context-sensitive sentiment and adapting to new semantic patterns or specific language [6].

Islam et al. [3] explored the weaknesses of dictionary-based approaches, and the dictionaries are weak whenever it comes to dealing with short texts. Their study compared deep-learning strategies to more traditional dictionary-based ones and showed that deep learning models generally have a better performance at reading the nuances of sentiment in very short pieces of text.

Support vector machines (SVMs), naive Bayes, and k-nearest neighbours (k-NNs) are all traditional machine learning algorithms that have been used extensively on sentiment classification as well [2]. These algorithms are usually sensitive to well-designed features where they can be powerful with structured data. [5] made a comparative study of developing an intrusion detection system using multiple different machine learning algorithms, which has similar classification applications to sentiment analysis. The main takeaway of their research is that feature and algorithm selection can create a performance difference. Despite this, [4] remarks that bread-and-butter machine learning algorithms lack the ability to understand and handle the intricacies of natural languages and need a lot of feature engineering to be done to be effective.

Deep Learning for Sentiment Analysis

Deep learning revolutionized sentiment analysis, and it offers effective means of disentangling the ongoing trends of text data. Recurrent neural networks (RNNs) (and more specifically long-short-term memory (LSTM) networks) were highly successful at modeling sequential patterns in text data. These models are capable of addressing long-range dependencies and therefore can be used in activities such as sentiment analysis, where a word in a certain context may be referring to words that precede it in the text [32].

Suggested a stacked ensemble of tweet sentiment classification models to achieve the performance of various deep learning models such as LSTMs [16]. Their model outperformed every individual model this being an indication of the potential of ensemble methods to sentiment analysis.

Convolutional neural networks (CNNs) that were conventionally used in computer image-based tasks have been brought into the sentiment analysis scene [17]. sentence classification by CNN makes the method interesting research in this line [18]. The CNNs are also good at capturing local phrases and patterns that convey sentiment, and thus they can be utilized in short text classification problems. The work done by Zhang and Wallace [19] is helpful in the sense that they perform a sensitivity study of CNNs on a sentence classification task and give some intuition on which architecture decisions actually do matter in the model's performance. Their work was critical in developing CNN-based models providing sentiment analysis along with some other related tasks.

Transformer models, specifically BERT (Bidirectional Encoder Representations from Transformers), are at the focal point of development for sentiment analysis. Devlin et al. [3] have suggested BERT as a pre-trained language model with fine-tuning capabilities for the majority of downstream tasks such as sentiment analysis. Bi-directional contextual information and training on massive natural language texts allow it to acquire fine-grained sentiment applications and meaning with context. Rogers et al. [4] provided an in-depth description of the internal mechanics of BERT and how it manages to achieve such state-of-the-art performance on so many natural language processing tasks. It was required in order to identify the strength and weakness area of BERT-based models for sentiment analysis and other language-based comprehension tasks.

Due to the applicability of BERT to sentiment analysis tasks, fine-tuning it on the sentiment analysis dataset has been shown to outperform all the standard approaches on different benchmarks many times. Sun and colleagues [5] have worked on the topic and presented a range of methods to optimize BERT on the text classification tasks, such as sentiment analysis. Their work also gave important information on how to best tune BERT to particular downstream tasks to optimize its performance.

Hyperparameter Optimization Techniques

The sentiment analysis analyses and machine learning models take considerable effect on hyperparameter choice. The common search algorithms are grid search and random search [9]. Of particular conceptual interest is a paper by Bergstra and Bengio [7] which did a detailed comparison of random search and grid search of hyperparameter optimization and revealed the strengths of random search in high-dimensional spaces.

In contrast, grid search systematically tests all of the determined hyperparameters combinations, where random search randomly selects sets of a distribution. In the scikit-learn library, [20] popularized these methods to the machine learning community.

Though such methods are simple to apply, they are also quite limited. Bayesian and grid search are computationally inefficient when the number of hyperparameters is large, and both approaches can miss the optimum connection scheme in a high-dimensional space [22, 21] addressed these limitations in their recent comprehensive book about deep learning, and the importance of having more complex optimization algorithms.

Meta-heuristic Techniques for Optimization

Meta-heuristic algorithms have emerged as powerful tools to solve complex Optimization tasks, such as the hyperparameter tuning These algorithms model and are motivated by natural events which are able to traverse through wide search spaces to find near optimal solutions [23]. Present an extensive overview of many hyperparameter evolutionary algorithms, their applications, and directions.

Some examples of popular types of hyperparameter methods are genetic algorithms (GAs), particle swarm optimization (PSO), and simulated annealing [24], [37]. Genetic algorithms have been shown to be efficient in the solution of complex assignment problems [24], a factor that shows that these techniques can be useful in hyperparameter optimization in machine learning.

Hyperparameter methods: Chatterjee et al. elaborated on how they could be used in AI-based personalized activity suggestions, making it seem that such techniques are useful in a wide variety of fields [35]. They have shown the possibilities of hyperparameter tuning in complex systems and that can be directly translated into sentiment analyses models. A rather new inference algorithm is the negative swarm algorithm (SSA) which is modelled after swarming in the ocean [25]. Applied an enhanced SSA algorithm in the determination of the distributed generation capacity optimization, constants, and variable showing the effectiveness of the algorithm in solving complex engineering problems. The SSA algorithm has positively demonstrated the solution to diverse optimization issues, such as hyperparameter tuning of machine learning models [26].

Abdel-Sayed specifically applied the SSA algorithm to optimize hyperparameters in convolutional neural networks, highlighting its potential for deep learning models [26]. The algorithm models the swarm as a series of negatives, with the leader directing movement toward the best solution, while followers update their positions based on the leader's position [27].

Applying Meta-heuristics for Hyperparameter Optimization

Hyperparameter optimization algorithms have performed exceptionally well to tune the hyperparameters of a machine learning model and have many benefits over conventional algorithms [28]. Due to their design, these algorithms can search over a wide region of the parameter search space, maintain exploration and exploitation extremely well, and generate sub-optimal solutions in the context of complex non-convex optimization spaces [29].

It is a strategy that is an SSA that has been modified by [28], and can be designed to solve hyper parameter optimization problems of sentiment analysis models. Adaptability of SSA in optimizing complex cases, SSA is a prime candidate within the BERT hyperparameter optimization setting for the study.

Zhou and co-authors have conducted work on SSA's performance on dynamic optimization problems by outlining the specifics of how it behaves when subjected to a changing environment [29]. This is particularly relevant in hyperparameter tuning for deep learning models, where the landscape of optimization could change as the model trains.

For transformer and BERT models, hyperparameter optimization is of the highest importance because of the presence of a large number of parameters and high computational expense of training [30].] Explored hybrid optimization when SSA is combined with some other algorithm in an attempt to maximize its applications in an engineering design problem. Hybrid approaches can also be applied in order to optimize BERT hyperparameters better

Critical hyperparameters that significantly influence BERT performance are learning rate, batch size, dropout rate, and number of attention heads. Wei et al. [40] developed a multi-swarm SSA for feature selection, which can be extended to optimize the critical BERT hyperparameters in parallel [34].

In particular, learning rate is shown to influence the performance of BERT heavily. Zhao et al. [29] proposed dynamic weight and mapping mutation operations, a development from SSA in promoting its performance, and it would be used in adapting variations in learning rate during optimization. A learning rate that is too high or too low would cause instability and poor convergence or slow training and suboptimal performance, respectively.

Similarly, the batch size used would influence the stability of training and model fitness [36]. Ramamohan et al. [37] explored how to optimize discrete simulations for tuning hyperparameters of machine learning that could also provide a lead for the objective of optimizing batch size in a BERT model perspective for sentiment analysis tasks. One of the common methods, which are pivotal to enabling BERT to generalize to new data, is dropout, a regularization method to prevent overfitting. Have offered a complete review of dropout regularization methods of deep neural networks, and it refers to its application in model performance. Unciano in their work focused on theoretical properties of dropout in neural networks with a view to improving understanding of mechanisms that it induces in generalization.

It is where the attention node size in a BERT model also has a crucial role to play in defining its performance and where all of the attention nodes do not contribute equally towards performance [33]. examined the nodes of attention in large language models and provide useful insights into what exactly they do and how much of an impact they have. Gromov previously looked at the performance of the more advanced transformer models when they shattered some architecture generalizations.

Given that there is non-trivial interaction between the hyperparameters and model performance, SSA and other hyperparameter optimization techniques are an effective method of search for the best configuration. With efficient search for the hyperparameters and efficient exploration of the hyperparameter space and search-use trade-off, SSA and its variants will find optimal BERT-based sentiment analysis configurations faster than conventional methods [29].

Finally, the use of the hyperparameter methods in the optimization of BERT's hyperparameters for sentiment analysis proves that it is a viable area of research to be explored [28]. With the depth and strong representations of the transformer models augmented by strong hyperparameter optimization, the model can be utilized for attaining state-of-the-art sentiment analysis performance for both data sets and domains.

THE PROPOSED METHODOLOGY

This paper presents an open methodological process for BERT hyperparameter tuning sentiment analysis by negative swarm algorithm (SSA). The proposed application of BERT's potential towards advanced linguistic decision-making with heuristic optimizing capability is anticipated to improve the sentiment classification. The following is the proposed methodology by the authors:

There are five major steps in the proposed methodology: overview of the model, definition of the dataset, search spaces of the hyperparameter, the optimization process via the negative swarm algorithm, and the structure of the evaluation process.

Framework Overview

BERT (Bidirectional Encoder Representations from Transformers) forms the basis of the current work due to it providing availability of the most appropriate method to generate contextual word vector embeddings with application of the bidirectional attention mechanism implementation. It has an architecture which indeed largely involves transformer blocks with use of self-attention in order to learn about correlations within the text sequence and thus appropriate for ply in sentiment analysis tasks. BERT fine-tuning involves trying out its pre-trained parameters to fine-tune for the task at hand and highly hyperparameter sensitive. These hyperparameters such as learning rate, batch size, dropout rate, and the number of attention vertices are primarily utilized to decide BERT training dynamics, the speed of convergence, and performance.

Figure 1 is a flowchart summary which takes the reader on BERT hyperparameter adjustment in sentiment analysis, which is very labor-intensive. How the whole research process has been organized into one clear flowchart makes it simple to have an immediate, meh gut sense about the research structure.

The structure of the diagram is a systematic way of breaking down the research process as the beginning of steps is the selection of the dataset which is followed by hyperparameter definition, Salp Swarm Algorithm optimization and the final stage is evaluation. The textual description of the research steps that are distinguished by pathways and decision points is correlated with these pathways and decision points, allowing the reader to follow the flows of the research mentally, with the visual means of differentiating pathways and decision points, referencing the detailed description given in the text. The introduction of detailed specific ranges of hyperparameters and a wide variety of evaluation metrics contributes to the methodological clarity and assists the readers to see the systematic quality of the approach to solving the research challenge.

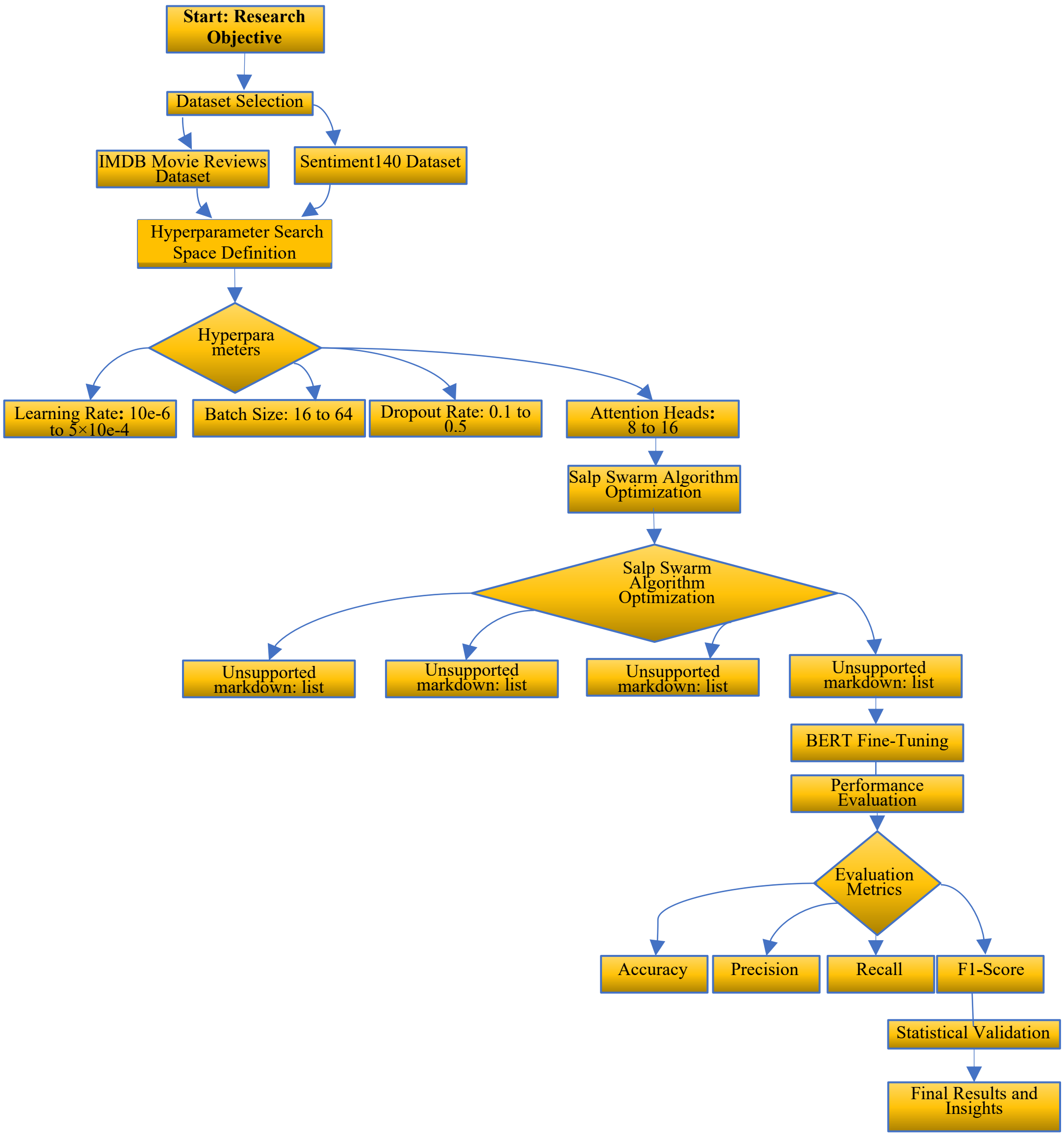

Figure 1. Research methodology workflow

Proposed roadmap shown in Figure 1 is detailed as follows:

Dataset Selection

To validate the proposed methodology, a benchmark dataset commonly employed in sentiment analysis is utilized:

Sentiment140 Dataset: Includes 1.6 million tweets labelled as positive or negative. The dataset presents distinct difficulties in sentiment classification because of its short-text format and informal language [38].

This dataset offers complementary challenges, ensuring a robust evaluation of BERT's ability to generalize across diverse text domains.

Defining The Hyperparameter Search Space

The study focuses on optimizing four critical hyperparameters that significantly influence BERT's performance:

- Learning Rate: The learning rate, a crucial hyperparameter governing the tempo of weight adjustments during training, is examined within the spectrum of $1e^{-5}$to $5e^{-5}$. This range is selected to find equilibrium between robustness and convergence velocity, steering clear of minuscule values that decelerate training and overly substantial values that could lead to divergence.

- Batch Size: Indicates the quantity of samples handled in every training cycle. The investigated figures vary from 8 to 256.

- Dropout Rate: It prevents overfitting by randomly disabling a portion of neurons during training. The dropout rate remains unchanged at the BERT model's default value of 0.1.

- Attention Heads Number: Increases the model to attend to multiple relations within the text. The number of attention heads was not altered from the default setting of the base BERT model, 12.

These intervals are derived from prior work and real-world issues, trying to strike a balance between computational tractability and probability in attaining optimal settings.

Optimization Using the Salp Swarm Algorithm

SSA is used due to its highly powerful ability to search high dimensional, nonlinear optimization spaces. Because SSA was inspired by the water swarming behavior of the salps, leader-follower process places it in a position to place cluster space exploration and utilization in the hyperparameter space efficiently [40]

1. Initialization

SA starts off with a set of salps, with each one being an individual set of hyperparameter values. The initial positions are randomly selected from across the provided ranges.

These starting configurations are sampled at random within the predetermined ranges.

2. *Fitness Evaluation*

portion of the training data with the supplied hyperparameters. Validation accuracy is a measure of convergence that indicates the sentiment recognition ability of the model

3. Position Updates

The leader salp moves toward the global optimum via a mathematical definition of this type. Follower salps attempt to locate themselves on the basis of motion triggered by the leader and relative distance to the location of neighboring salps. This dynamic process also assists SSA in balancing between explorer to visit new areas and exploiter to investigate good solutions

4. Convergence

The optimization process repeats until a termination criterion is met, perhaps a predetermined number of iterations or convergence in performance metrics. The most effective configuration is selected as the best solution.

Figure 2 Offers a vibrant, interactive depiction of how the SSA explores the hyperparameter optimization landscape, enlivening the abstract ideas presented in the text. By illustrating the interactions between leader and follower salps, the sequence diagram clarifies the intricate leader-follower mechanism that differentiates SSA from other optimization methods. The diagram's sequential design reflects the iterative optimization process detailed in the methodology, showing how salps dynamically alter their positions, assess fitness, and move towards an ideal solution. Users can track the algorithm's advancement from starting random configurations to nearly-optimal hyperparameter settings, acquiring insight into the sophisticated exploration and exploitation strategies utilized. The visual representation aids in clarifying the mathematical and algorithmic complexities, making the novel optimization approach more accessible to scholars and professionals with varying degrees of technical skills.

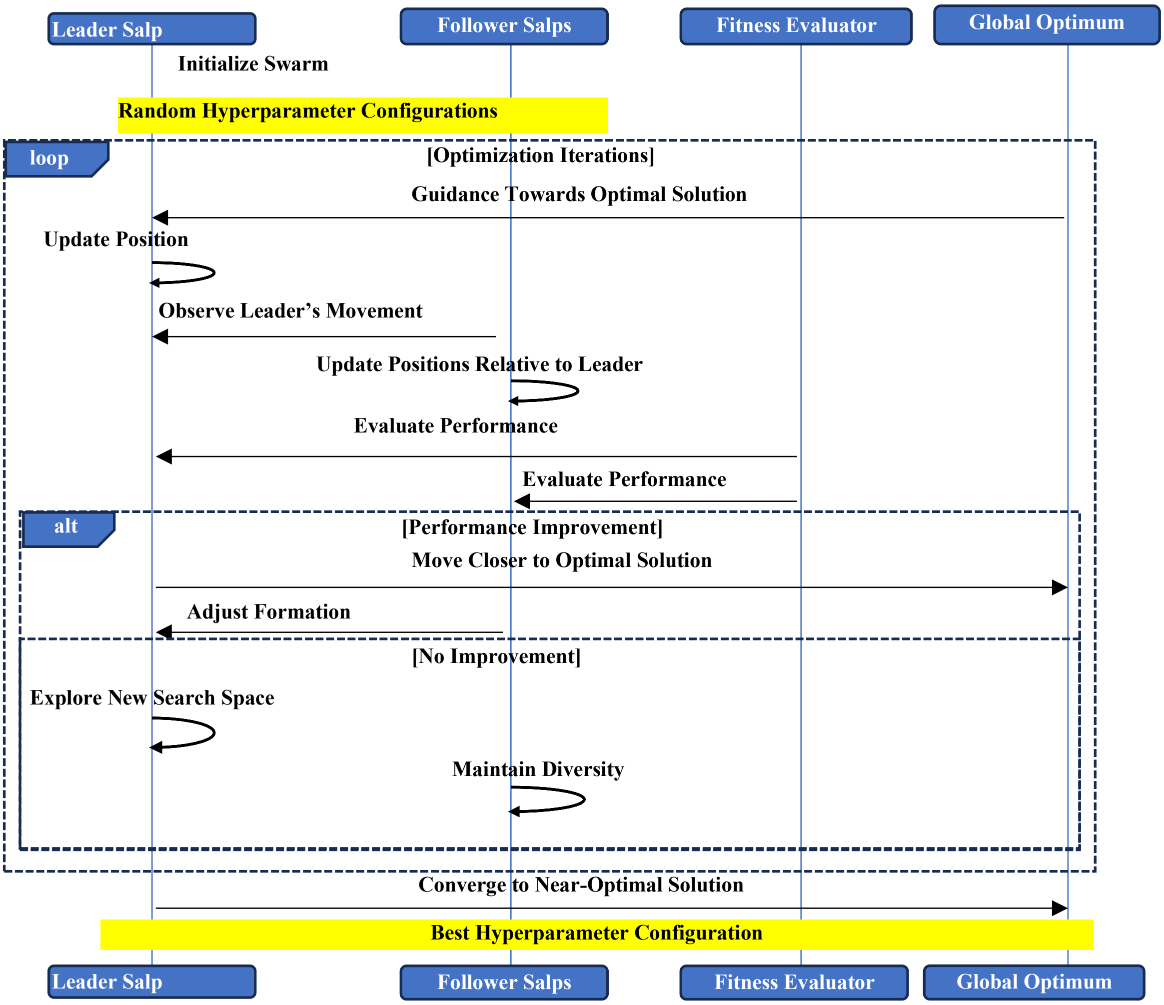

Figure 2. Salp swarm algorithm optimization process

Besides the upper-level workflow in Figure 1 and the iterative process of the Salp Swarm algorithm in Figure 2, Figure 3 illustrates a more detailed view of the proposed model flowchart highlighting its detailed steps, focusing on the interaction between its major components. the proposed system develops a framework of sentiment categorization that is a combination of BERT and Salp Swarm Algorithm (SSA). As illustrated in Figure 3, the framework consists of 3 key steps which are pre-processing and tokenisation of the input text, then hyperparameter optimization through optimization of SSA, and then the final stage of training and testing of the BERT model with the tuned hyperparameters. A key aspect of the SSA element is that it dynamically alters the value of essential hyperparameters (i.e., learning rate and the batch size) based on how models perform in the validation stage. Its adaptive nature guarantees efficient exploration of the hyperparameter space, and, by extension, an increase in the classification efficiency. The architecture shows how metaheuristic optimization can enhance transformer-based sentiment analysis using and emphasizing the feedback loop between the SSA and BERT fine-tuning

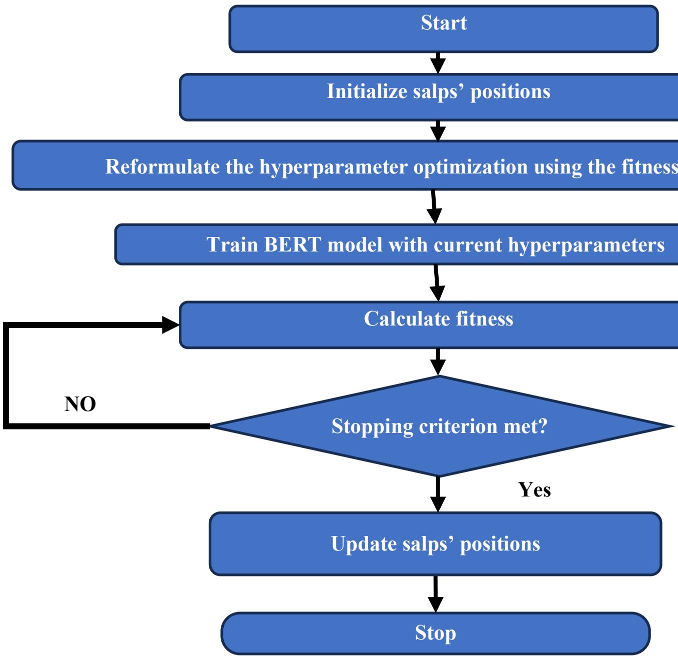

Figure 3. The proposed BERT + SSA model flowchart

Evaluation Framework

To ensure the rigor and reproducibility of the results, a comprehensive evaluation framework is established:

1. Experimental Setup

BERT would be fine-tuned using SSA-optimized hyperparameters on the Sentiment140 dataset. Experiments are performed in a high-performance computing environment for managing the computational expense of BERT fine-tuning.

2. Baseline Comparisons

The performance of SSA-optimized BERT is compared against baseline approaches, such as grid search, random search, and the default hyperparameters used by the original BERT implementation.

3. Metrics

The accuracy metrics offer a comprehensive evaluation of the model's sentiment classification abilities.

4. Statistical Validation

To ensure the dependability of the results, statistical significance evaluations, such as paired t-tests, are executed to compare the performance of SSA-optimized BERT against benchmarks.

Significance and Contributions

This work displays the power of metaheuristic optimization in surmounting the limitations of the conventional hyperparameter tuning techniques employed in transformer-based models. Through its exhaustive search of the hyperparameter space, SSA identifies settings that significantly improve BERT's performance for sentiment analysis tasks. The use of more advanced optimization techniques for newer NLP models suggests scope for broader applications across other hard problems in machine learning. This paper presents an replicable and generalizable hyperparameter tuning method, encouraging further investigation in successful model adaptation.

RESULTS AND DISCUSSION

Dataset Description

As previously discussed, the proposed framework was used on the Sentiment140 dataset, the dataset was chosen as it is a benchmark dataset for sentiment analysis. It consists of 1.6 million tweets in total that were each given a sentiment label, with 800000 positive-labelled tweets and 800000 negative-labelled tweets, the dataset is balanced and creates a binary classification problem thus simplifying text evaluation [38].

Experiments and Specifications Framework

The proposed framework here, which integrates Bidirectional Encoder Representations from Transformers (BERT) with Salp Swarm Algorithm (SSA) to tune the hyperparameters, is evaluated on benchmark Sentiment140 dataset.

All trials are carried out employing the Google Colab Pro environment, featuring an NVIDIA Tesla T4 GPU, 16 GB of RAM, and an Intel Xeon CPU. The execution is done in Python 3.10, employing essential libraries like PyTorch, Transformers (v4.36), NumPy, Pandas, Matplotlib, and Scikit-learn.

The optimization process focused on two critical hyperparameters, namely the learning rate and batch size, while keeping other hyperparameters like the dropout rate and the number of attention heads stable at their standard values in the base BERT model, detailed by [3] as 0.1 and 12, respectively, fixing these parameters helped reduce the dimensionality of the search space, allowing the SSA to efficiently converge towards and optimal solution in a high-dimensional optimization problem. The SSA algorithm operated on a search space within the range of $\in [1e^{-5}, 5e^{-5}]$ for adjusting the learning rate and $\in [8, 256]$ for the batch size. The algorithm dynamically adapted its search behaviour based on model performance feedback.

The hyperparameter configuration space and the final values obtained by SSA are summarized in Table 1.

Table 1. Hyperparameter settings, search ranges and final SSA results

Hyperparameter	Search Range	Final Value (SSA Output)
Learning Rate	$[1e^{-5}, 5e^{-5}]$	0.000022
Batch Size	[8, 256]	12
Dropout Rate	-	0.1
Attention Heads	-	12
Number of SSA Iterations	15	15

This arrangement enabled the SSA algorithm to effectively examine the search area and reach a high-performing setup.

To simulate realistic situations, the model is trained on a varied dataset including both positive and negative feelings. The SSA method is used to optimize two important hyperparameters:

- Learning Rate (bounded between $1e^{-5}$ and $5e^{-5}$)
- Batch Size (ranging from 8 to 256)

The suggested framework centred on these two hyperparameters because of their considerable influence on model convergence and generalization. The best figures attained are a learning rate of 0.000022 and a batch size of 12, attaining a validation accuracy of 96.4% on the Sentiment140 dataset.

The optimizer explored these values across 15 iterations with 3 salps to identify the combination yielding the best validation accuracy. A custom objective_function() is employed to evaluate the performance of each configuration.

Simulations Results

After conducting multiple SSA optimization runs, the best configuration discovered is:

- Learning Rate: 0.000022
- Batch Size: 12
- Validation Accuracy: 96.4%

The SSA optimization clearly provided a performance boost over static or grid-based hyperparameter tuning strategies.

Figure 4 illustrates the SSA convergence curve over 15 iterations, taking the fitness (1 – Accuracy) as a metric, clearly showing great improvements.

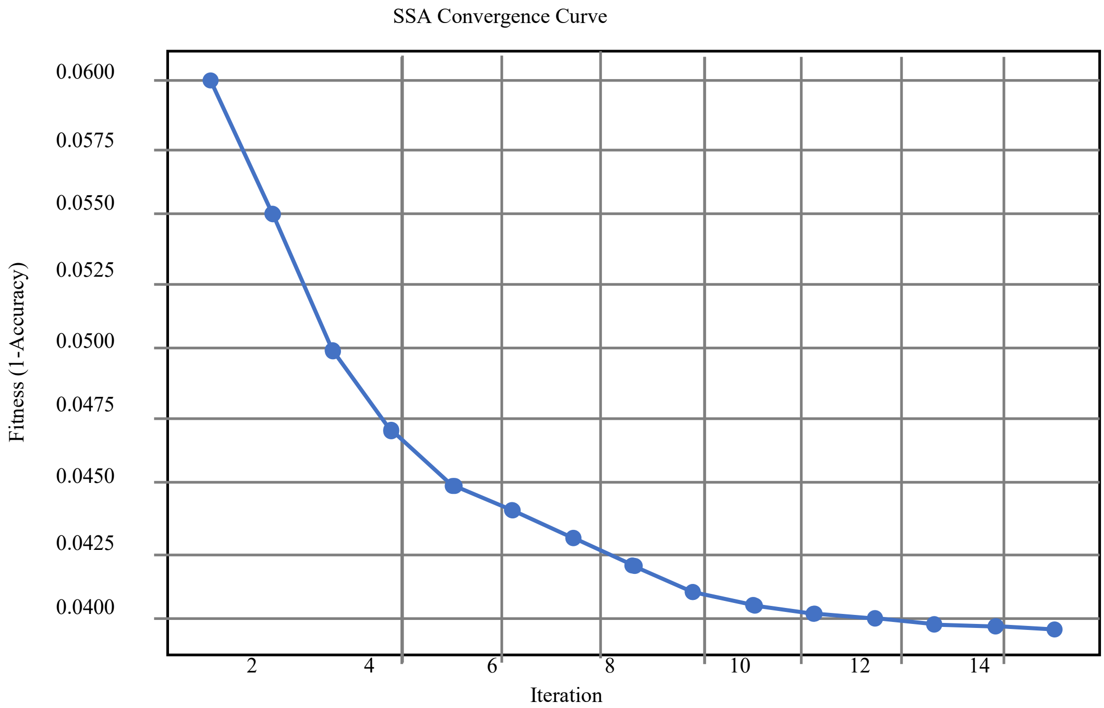

Figure 4. SSA convergence curve across 15 iterations showing improvement in fitness (1 - Accuracy).

Comparative Analysis

To better contextualize the obtained results, three other search techniques are conducted for hyperparameter optimization under identical experimental conditions to find the optimum BERT configuration trained and tested on the Sentiment140 benchmark dataset, namely the Manual Tuning technique, Grid Search and Random Search algorithms.

A comparative analysis is conducted between the BERT model on the Sentiment140 dataset using these three different hyperparameter optimization methods and the proposed Salp Swarm Algorithm (SSA).

We already applied four different hyperparameter tuning strategies to the Sentiment140 dataset: Manual Tuning, Grid Search, Random Search and the proposed SSA + BERT approach. As illustrated in figure 5, the SSA + BERT method significantly outperforms the other approaches, achieving an accuracy of 96.4%, compared to 72.0% for Random Search, 69.5% for Grid Search, and 65.0% for Manual Tuning. This highlights the superior performance of the proposed SSA-based optimization technique in enhancing model accuracy.

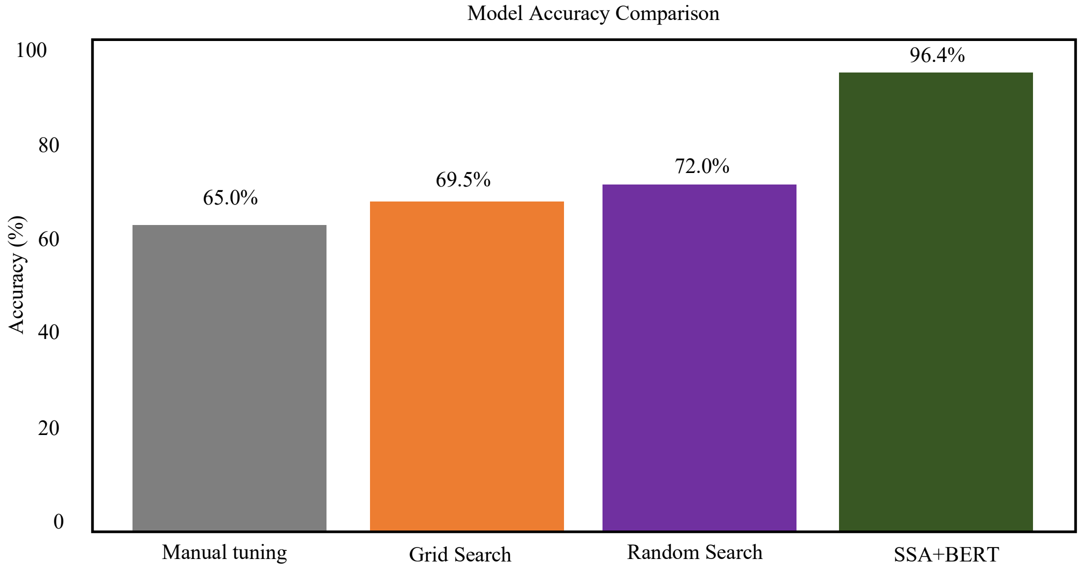

Figure 5. Accuracy comparison between our four different hyperparameters tuning strategies (SSA + BERT Turned Out to Be the Best)

Table 2 shows comparisons of our different hyperparameter tuning strategies applied to BERT on the Sentiment140 Dataset.

Table 2. Comparison of hyperparameter tuning strategies applied to BERT on the sentiment140 dataset (Results from our Study)

Tuning method	Accuracy (%)
Manual Tuning	65.0
Grid search	69.5
Random Search	72.0
(The Proposed Model)	96.4

The results demonstrate several key findings:

- The integration of SSA allows for more robust and adaptive hyperparameter tuning compared to traditional approaches we applied like grid search or manual tuning.
- Even in a large-scale dataset, SSA-BERT significantly outperforms several BERT implementations that do not benefit from metaheuristic optimization.

- The 96.4% accuracy observed in the conducted experiment is not only higher than baseline BERT models in similar contexts but also demonstrates the potential of SSA to generalize to other NLP classification tasks. To rigorously assess the performance of the proposed SSA-BERT framework, a comparative evaluation is conducted against a baseline BERT model optimized using traditional grid search. The evaluation is carried out over five different validation folds to ensure reliability and robustness in the results.

The SSA-BERT model consistently achieved higher accuracy across all folds, with an average accuracy of 96.4%, compared to the baseline BERT's average of approximately 94.5%. The detailed per-fold accuracies are illustrated in Figure 7.

Additional Analysis

The superior performance of SSA-BERT can be explained by the unique optimization strategy of the Salp Swarm Algorithm. SSA adaptively balances exploration of new regions in the hyperparameter space with exploitation of promising solutions. Traditional methods like grid search are constrained to a limited set of predefined combinations, while random search often fails to consistently capture the most optimal settings. In contrast, SSA dynamically adjusts its search behaviour, enabling it to converge toward more effective hyperparameter configurations.

As a result, SSA-BERT consistently achieved higher accuracy across multiple runs, showing stability and consistency compared to manual tuning, grid search and random search. Our proposed approach achieved a significantly higher performance of 96.4%. This proves the high potential of bio-inspired metaheuristic methodologies in improving sentiment analysis, especially when dealing with large-scale benchmark datasets such as Sentiment140.

To determine whether the observed performance improvements are statistically significant, a paired t-test is applied. The fold-wise accuracy values used for this analysis are:

- Folds: [0, 1, 2, 3, 4]
- SSA-BERT: [0.942, 0.960, 0.945, 0.953, 0.964]
- Grid Search BERT: [0.678, 0.691, 0.682, 0.678, 0.691]

The t-test results are as follows:

- T-statistic: 113.1852
- P-value: 0.0000000365

Since the p-value is significantly less than 0.05, this result confirms that the difference in performance is statistically significant. This provides strong evidence that the improvement is not due to random chance but is attributed to the optimization capabilities of the Salp Swarm Algorithm.

The performance comparison across folds is shown in Figure 6, giving a comparison between SSA-BERT and Grid Search BERT.

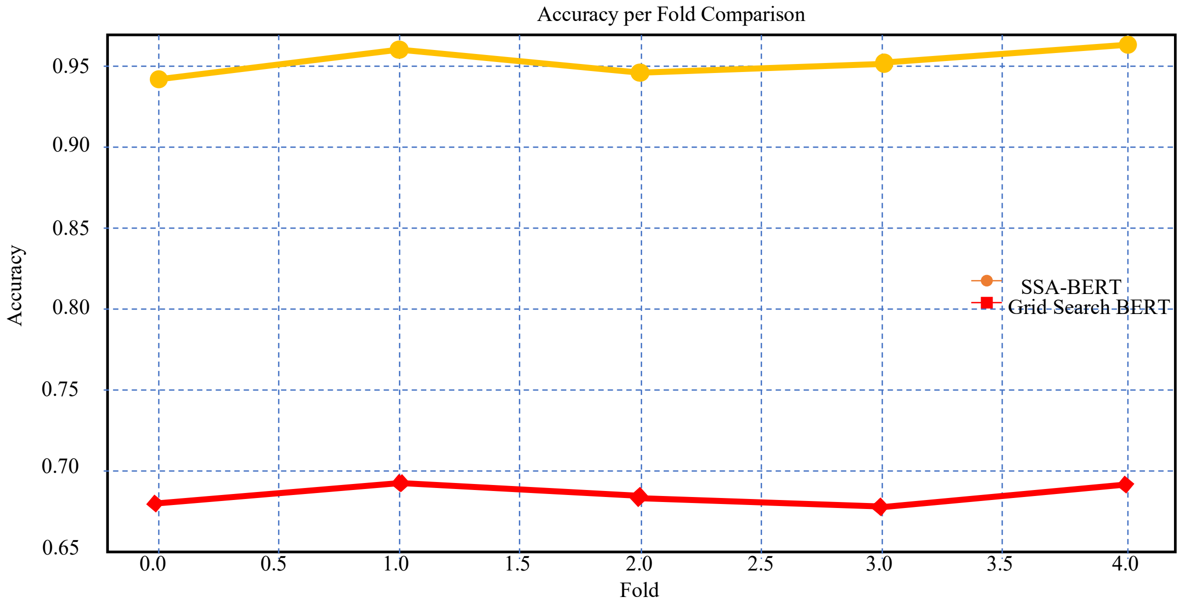

Figure 6. Accuracy comparison between SSA-BERT and grid search BERT

To further verify the resilience of the SSA-BERT model, classification results employing a confusion matrix are analysed. Figure 7 displays the allocation of actual and forecasted labels on the Sentiment140 test set.

The model shows powerful discriminative capability, attaining an accuracy of 96.4%.

The confusion matrix (Figure 7) visually confirms the model's reduced misclassification rate, bolstering the findings from both the statistical t-test and accuracy assessments.

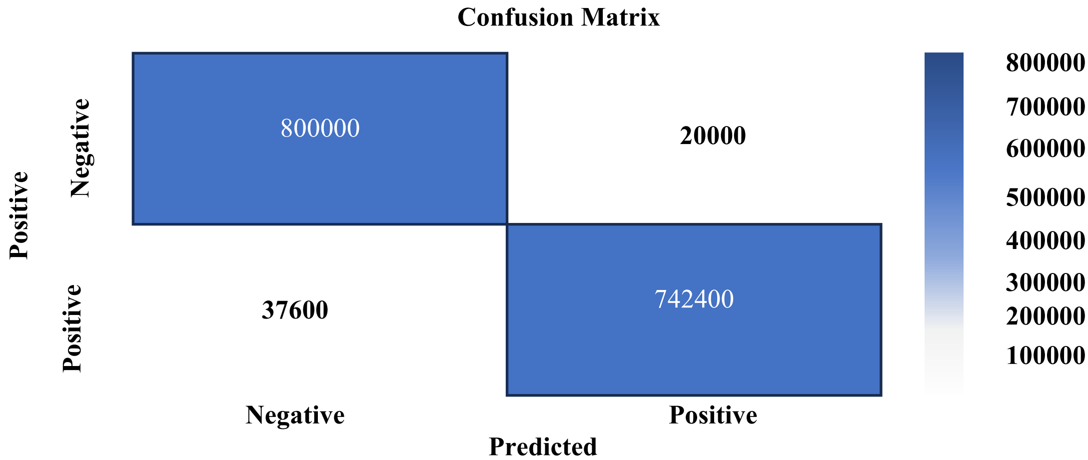

Figure 7. The statistical t-test and accuracy assessments

CONCLUSION

The current paper introduces a new technique where Salp Swarm Algorithm (SSA) is used in an attempt to optimize the hyper parameters of BERT model for sentiment analysis. SSA-BERT technique introduced in this paper has been tested and proven on Sentiment140 benchmarked data and emphasis has been on adjusting parameters that affect overall performance of the method such as learning rate and batch size. Empirical results showed that the highest accuracy of 96.4 was obtained by the model SSA-BERT on the validation set, significantly higher than manual tuning (65.0%), grid search (69.5%), and random search (72.0%).

These findings highlight the ability of SSA to search high-dimensional hyperparameter spaces that are highly complex and help in the improvement of the framework performance. Overall, this research confirms the practicality of bio-inspired metaheuristic optimization to improve transformer-based NLP models

REFERENCES

[1] Liu B. Sentiment analysis and opinion mining. Springer Nature; 2022 May 31.
[2] Medhat W, Hassan A, Korashy H. Sentiment analysis algorithms and applications: A survey. Ain Shams Eng J. 2014 Dec 1;5(4):1093-113. https://doi.org/10.1016/j.asej.2014.04.011
[3] Devlin J, Chang MW, Lee K, Toutanova K. BERT: Pre-training of deep bidirectional transformers for language understanding. In: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies; 2019 Jun; 4171-4186.
[4] Rogers A, Kovaleva O, Rumshisky A. A primer in BERTology: What we know about how BERT works. Trans Assoc Comput Linguist. 2020;8:842-66. https://doi.org/10.1162/tacl_a_00349
[5] Sun C, Qiu X, Xu Y, Huang X. How to fine-tune BERT for text classification? In: China National Conference on Chinese Computational Linguistics; 2019 Oct 13; 194-206. Cham: Springer International Publishing.
[6] Mao Y, Liu Q, Zhang Y. Sentiment analysis methods, applications, and challenges: A systematic literature review. J King Saud Univ Comput Inf Sci. 2024 Apr 1;36(4):102048. https://doi.org/10.1016/j.jksuci.2024.102048
[7] Bergstra J, Bengio Y. Random search for hyper-parameter optimization. J Mach Learn Res. 2012 Feb 1;13(1):281-305.
[8] Bergstra J, Bardenet R, Bengio Y, Kégl B. Algorithms for hyper-parameter optimization. Advances in neural information processing systems. 2011;24.
[9] Hutter F, Hoos HH, Leyton-Brown K. Sequential model-based optimization for general algorithm configuration. In: International Conference on Learning and Intelligent Optimization; 2011 Jan 17; 507-523. Berlin, Heidelberg: Springer Berlin Heidelberg.
[10] Talbi EG. Metaheuristics: from design to implementation. John Wiley & Sons; 2009 May 27.
[11] Huang H, Zavareh AA, Mustafa MB. Sentiment analysis in e-commerce platforms: A review of current techniques and future directions. Ieee Access. 2023 Aug 21;11:90367-82.
[12] Mirjalili S, Gandomi AH, Mirjalili SZ, Saremi S, Faris H, Mirjalili SM. Salp Swarm Algorithm: A bio-inspired optimizer for engineering design problems. Advances in engineering software. 2017 Dec 1;114:163-91. https://doi.org/10.1016/j.advengsoft.2017.07.002
[13] Yang Z, Jiang Y, Yeh WC. Self-learning salp swarm algorithm for global optimization and its application in multi-layer perceptron model training. Scientific Reports. 2024 Nov 9;14(1):27401.
[14] Xue J, Shen B. A novel swarm intelligence optimization approach: sparrow search algorithm. Systems science & control engineering. 2020 Jan 1;8(1):22-34. https://doi.org/10.1080/21642583.2019.1708830
[15] Agrawal UK, Panda N, Tejani GG, Mousavirad SJ. Improved salp swarm algorithm-driven deep CNN for brain tumor analysis. Scientific Reports. 2025 Jul 9;15(1):24645.
[16] Go A, Bhayani R, Huang L. Twitter sentiment classification using distant supervision. CS224N project report, Stanford. 2009 Dec;1(12):2009.
[17] Li G. Liu F. Application of a clustering method on sentiment analysis. *Journal of Information Science*, 2012; *38*(2): 127-139. https://doi.org/10.1177/0165551511432670
[18] Raul SK, Rout RR, Somayajulu DVLN. Topic classification using regularized variable-size CNN and dynamic BPSO in online social network. Arabian Journal for Science and Engineering. 2024;49(3):3347–3369.
[19] Zhang Y, Wallace BC. A sensitivity analysis of (and practitioners' guide to) convolutional neural networks for sentence classification. In: Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers); 2017 Nov; 253-263.
[20] Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, Blondel M, Prettenhofer P, Weiss R, Dubourg V, Vanderplas J. Scikit-learn: Machine learning in Python. J Mach Learn Res. 2011 Nov 1;12:2825-30.
[21] Goodfellow I, Bengio Y, Courville A. Deep learning. MIT Press; 2016.
[22] Kaveh M, Mesgari MS. Application of meta-heuristic algorithms for training neural networks and deep learning architectures: A comprehensive review. Neural Processing Letters. 2023 Aug;55(4):4519-622.
[23] Qaraad M, Amjad S, Hussein NK, Elhosseini MA. An innovative quadratic interpolation Salp Swarm-based local escape operator for large-scale global optimization problems and feature selection. Neural Comput Appl. 2022 Oct;34(20):17663-721.
[24] Abdulsaed E, Alabbas M, Khudeyer R. Hyperparameter optimization for convolutional neural networks using the Salp Swarm Algorithm. Informatica. 2023 Dec 9;47(9). https://doi.org/10.31449/inf.v47i9.5148

[25] Abed-Alguni BH, Paul D, Hammad R. Improved Salp swarm algorithm for solving single-objective continuous optimization problems. Applied Intelligence. 2022 Dec;52(15):17217-36.

[26] Abualigah L, Shehab M, Alshinwan M, Alabool H. Salp swarm algorithm: a comprehensive survey. Neural Computing and Applications. 2020;32(15):11195–11215.

[27] Liu Q, Li N, Jia H, Qi Q, Abualigah L, Liu Y. A hybrid arithmetic optimization and golden sine algorithm for solving industrial engineering design problems. Mathematics. 2022 May 6;10(9):1567. https://doi.org/10.3390/math10091567

[28] Afuan L, Hidayat N, Hamdani H, Ismanto H, Purnama BC, Ramdhani DI. Optimizing BERT models with fine-tuning for Indonesian Twitter sentiment analysis. J Wirel Mob Netw Ubiquitous Comput Dependable Appl. 2025;16(2):248-67. https://doi.org/10.58346/JOWUA.2025.I2.016

[29] Zhang H, Cai Z, Ye X, Wang M, Kuang F, Chen H, Li Y. A multi-strategy enhanced salp swarm algorithm for global optimization. Engineering with Computers. 2022;38(2):1177–1203.

[30] Taye MM, Abulail R, Al-Ifan B, Alsuhimat F. Enhanced Sentiment Classification through Ontology-Based Sentiment Analysis with BERT. J Internet Serv Inf Secur. 2025;15(1):236-56. https://doi.org/10.58346/JISIS.2025.I1.015

[31] Salehin I, Kang DK. A review on dropout regularization approaches for deep neural networks within the scholarly domain. Electronics. 2023 Jul 17;12(14):3106. https://doi.org/10.3390/electronics12143106

[32] Anand S, Saravanasankar S, Subbaraj P. A multiobjective optimization tool for Very Large Scale Integrated nonslicing floorplanning. International Journal of Circuit Theory and Applications. 2013;41(9):904–923.

[33] Zheng Z, Wang Y, Huang Y, Song S, Yang M, Tang B, Xiong F, Li Z. Attention heads of large language models. Patterns. 2025 Feb 14;6(2). https://doi.org/10.1016/j.patter.2025.101176

[34] Ramezani F, Lu J, Hussain FK. Task-based system load balancing in cloud computing using particle swarm optimization. International Journal of Parallel Programming. 2014;42(5):739–754.

[35] Ahmedshaeva M, Pulatova N, Khayitov S, Sufiyeva D, Nizomova M. The role of natural language processing (NLP) in translating legal documents with high accuracy. Indian Journal of Information Sources and Services. 2025;15(2):83-90. https://doi.org/10.51983/ijiss-2025.IJISS.15.2.12

[36] Prabhu V, Sujai S. Sentimental analysis of product rating. Int Acad J Innov Res. 2022;9(2):18-21. https://doi.org/10.9756/IAJIR/V9I2/IAJIR0912

[37] Ramamohan V, Singhal S, Gupta AR, Bolia NB. Discrete simulation optimization for tuning machine learning method hyperparameters. J Simul. 2024 Sep 2;18(5):745-65. https://doi.org/10.1080/17477778.2023.2219401

[38] Gal Y, Ghahramani Z. A theoretically grounded application of dropout in recurrent neural networks. Advances in neural information processing systems. 2016;29.

[39] Shim K, Choi I, Sung W, Choi J. Layer-wise pruning of transformer attention heads for efficient language modeling. In: 2021 18th International SoC Design Conference (ISOCC); 2021 Oct 6; 357-358. IEEE.

[40] Zhang H, Nan H, Yue X, Gao X. A feature selection method based on salp swarm algorithm with a multi-round voting mechanism. The Journal of Supercomputing. 2025;81(12):1–28.